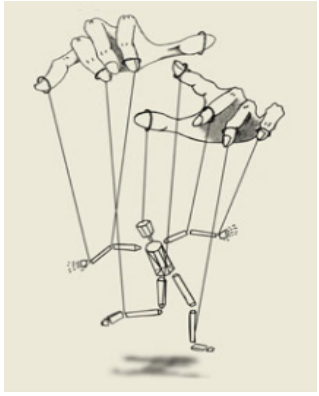


System Automation mit Puppet und Foreman

Toni Schmidbauer

9. Mai 2014

whoami

- SysAdmin@s-itsolutions
- toni@stderr.at
- `http://stderr.at`
- `http://github.com/tosmi`
- stderr@jabber.org

Agenda

- Kurze Umfrage
- Was ist Puppet?
- Was ist Foreman?
- Puppet@s-iTSolutions
- Was haben wir geplant?

Umfrage

Was ist Puppet?

- Declarative programming: telling the machine what you would like to happen, and let the computer figure out how to do it.
- Imperative programming: telling the machine how to do something

```
class linuxwochen2014 (
  $ensure = present
) {
  user { 'toni':
    ensure => $ensure,
    uid    => 4711,
    gid    => 100,
  }

  package { 'emacs-nox':
    ensure => installed
  } ->
  package { 'vim-enhanced':
    ensure => absent,
  }
}
```

Zuordnung von Klassen

- über /etc/puppet/manifests/site.pp

```
2
        node /^(foo|bar)\.linuxwochen\.at$/ {
          include linuxwochen2014
        }
```

- über Hiera (hiera_include('classes',[“”]))
- über einen External Node Classifier (Foreman)

Node classification

```
---
classes:
  black:
  puppetmgmt:
    manage_config: true
    puppetmaster: puppetd
    puppet_version: 3.2.4
  sudo:
    config_dir: /etc/sudoers.d/
    config_file: /etc/sudoers
    config_file_replace: false
    package: sudo
    package_source: ''
    purge: false
```

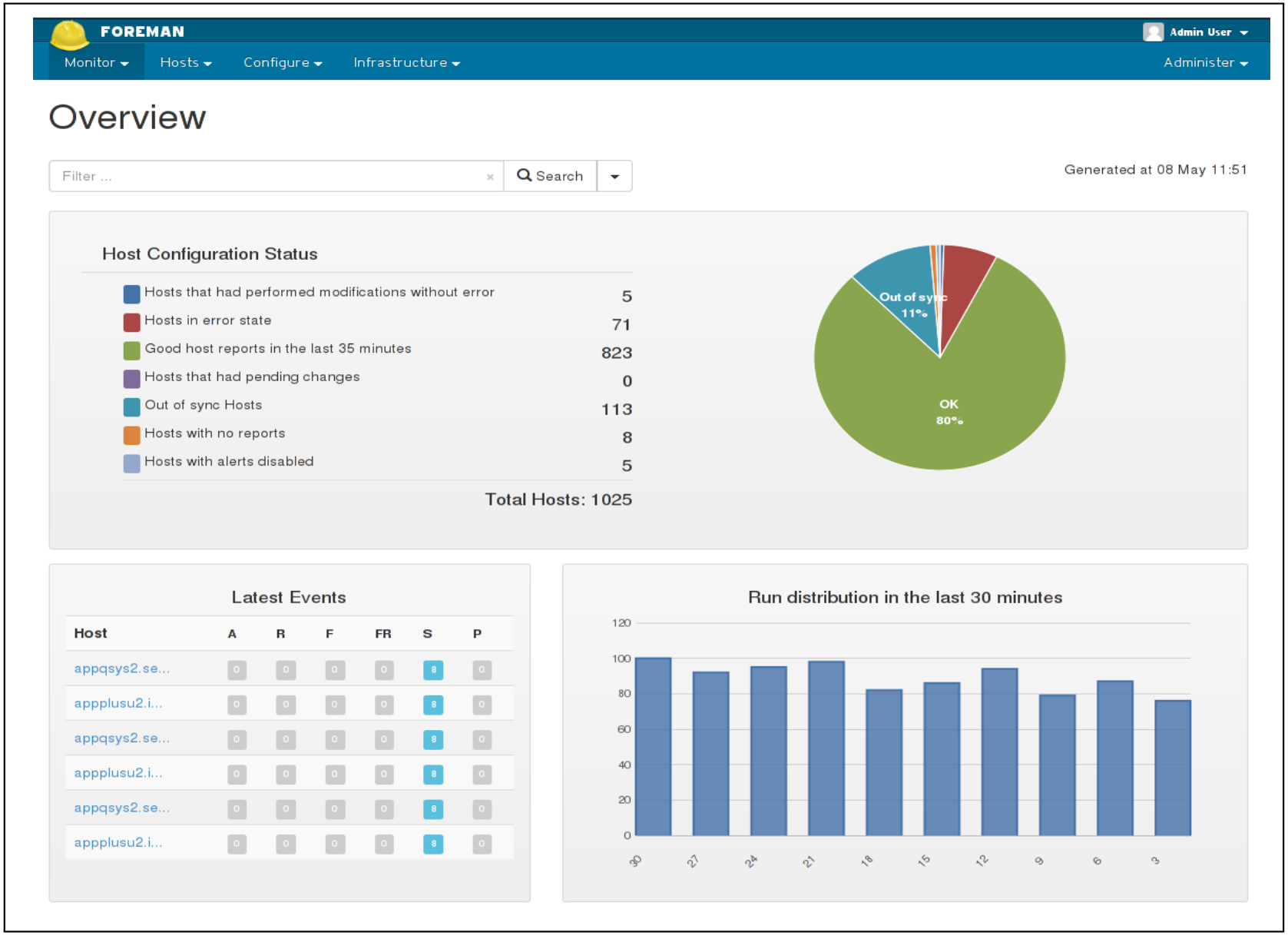
Puppet Environments

- Environments sind unabhängige Puppet Umgebungen
- Ein Master kann mehrere Environments zur Verfügung stellen

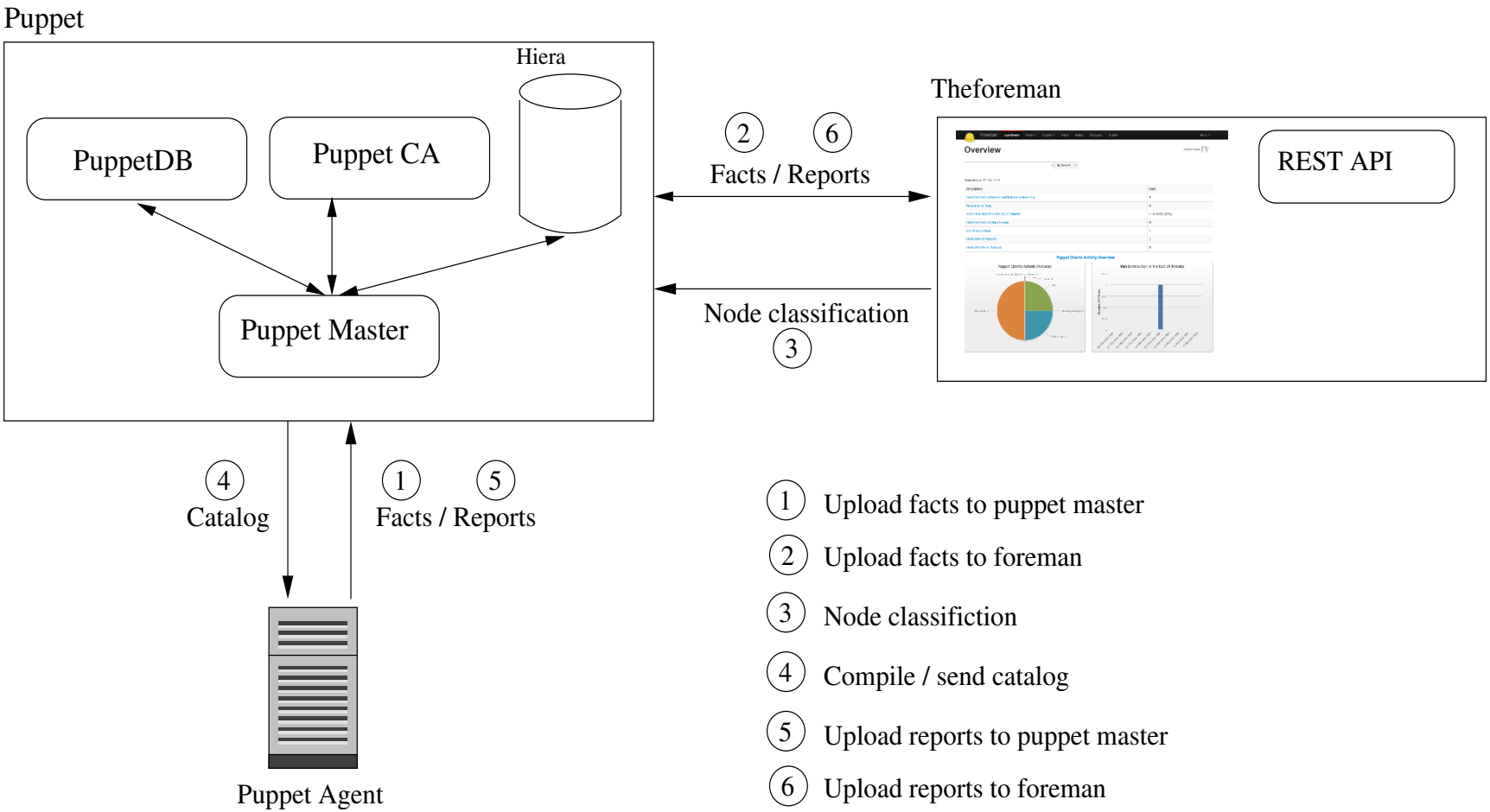
```
/etc/puppet/modules/...
/etc/puppet/manifests/site.pp

/etc/puppet/environments/linuxwochen/modules
/etc/puppet/environments/linuxwochen/manifests/site.pp
```

Was ist Foreman?



Puppet run


Puppet
Hiera
PuppetDB
Puppet CA
Puppet Master
2
6
Facts / Reports
Node classification
3
Theforeman
Overview
REST API
4
Catalog
1
5
Facts / Reports
Puppet Agent
1 Upload facts to puppet master
2 Upload facts to foreman
3 Node classifiction
4 Compile / send catalog
5 Upload reports to puppet master
6 Upload reports to foreman

Und jetzt?

Good

- Wie soll eine Entwicklungsumgebung aussehen?
- Wie testen wir den Puppet Code?
- Wie verwalten wir unseren Puppet Code?
- Wie soll unsere Puppet Umgebung aussehen?
- Wie erfolgt das Deployment des Codes?
- Wie verwalten wir Module von PuppetForge?

Wie soll eine Entwicklungsumgebung aussehen?

Vagrant

- `http://vagrantup.com`
- Ermöglicht virtuelle Entwicklungsumgebungen
- Vagrant Box ist ein vorkonfiguriertes Image
- Default VirtualBox andere Provider via Plugins (VMWare, KVM)

Demo

Wie testen wir den Puppet Code?

rspec-puppet

- Ruby RSpec Tests für Puppet
- Jedes Module muss RSpec Tests mitbringen

```
require 'spec_helper'
describe 'linuxwochen2014' do
  let :facts { { :osfamily => 'RedHat' } }

  context 'ensure is set to absent' do
    let :params { { :ensure => 'absent'} }

    it do
      should contain_user('toni').with({
                                         'ensure' => 'absent',
                                         'uid'    => '4711',
                                         'gid'    => '100',
                                       })
    end

    it { should contain_package('emacs-nox').with_ensure('installed') }
    it { should contain_package('vim-enhanced').with_ensure('absent') }
    it { should contain_package('emacs-nox).that_comes_before('Package[vim-enhanced]') }
  end
end
```

Demo

Wie verwalten wir unseren Puppet Code?

GIT

- Ein zentrales GIT Repository
- 3 Hauptbranches
  - Master
  - Testing
  - Production
- Feature Branches für neue Module
- Berechtigungssystem mit Gitolite

Wie soll unsere Puppet Umgebung aussehen?

Wie erfolgt das Deployment des Codes?

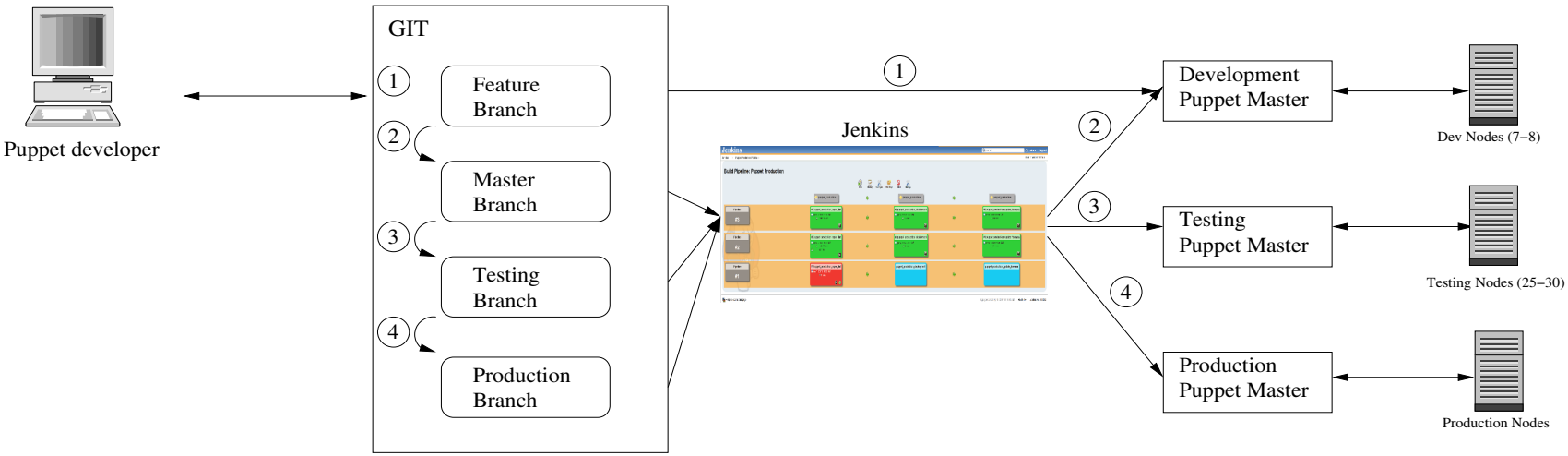
Umgebungen

- Development (Master)
  - Test auf 7 Entwicklungsserver
  - Push auf Remote Master Branch löst Deployment aus
  - RedHat 5,6 / Solaris sparc,i386 10,11 / AIX
- Testing
  - ca. 30 “Produktions” Server
  - Annotated Tags werden automatisch deployed (test_*)
- Production
  - ca 1000 Hosts
  - Annotated Tags werden automatisch deployed (prod_*)

Jenkins CI

- Continuous Integration Server
- Führt Jobs aus (z.b. Shell Scripts)
- Monitored und liefert Status Jobs
- Es können Abhängigkeiten zwischen Jobs definiert werden

Puppet Umgebung und Deployment



(1) Features Branches get automatically created on Puppet Master (Dynamic Environments)

(2) Master Branch gets deployed on commit via Jenkins

(3) Testing Branch gets deployed via GIT tag (test_*)
a normal commit to the Testing branch only runs tests

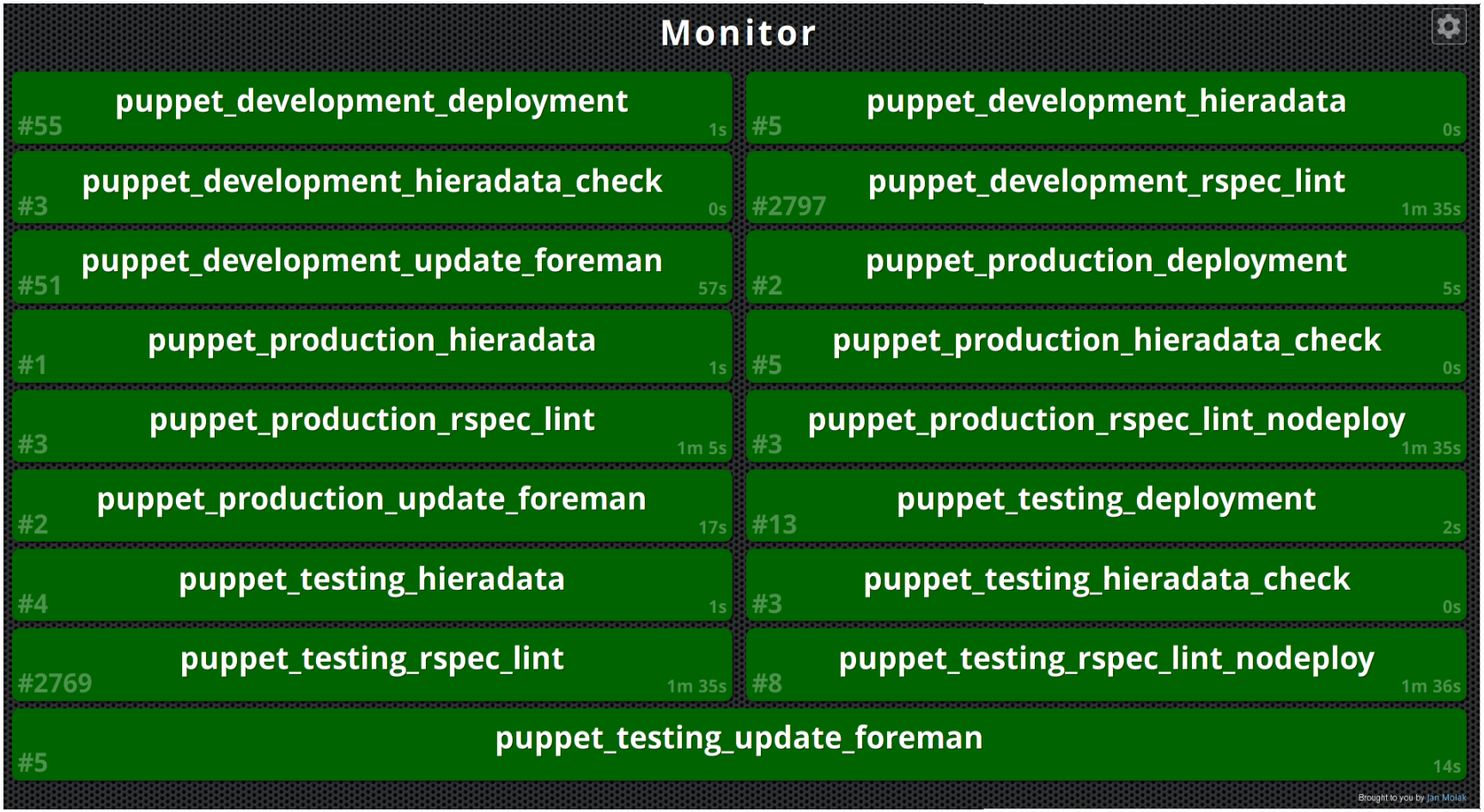
(4) Production Branch gets deployed via GIT tag (prod_*)
a normal commit to the Prodcution branch only runs tests

It's all the same for Hiera yaml files, except dynamic environments!

Deployment

Build Pipeline: Puppet Production
Run
History
Configure
Add Step
Delete
Manage
puppet_production...
puppet_production...
puppet_production...
Pipeline #14
GITTAG : prod_20140429_1
#14 puppet_production_rspec_lint
Apr 29, 2014 3:55:43 PM
1 min 39 sec
#12 puppet_production_deployment
Apr 29, 2014 3:57:28 PM
10 sec
#12 puppet_production_update_foreman
Apr 29, 2014 3:57:43 PM
13 sec
Pipeline #13
GITTAG : prod_20140425_1
#13 puppet_production_rspec_lint
Apr 25, 2014 2:58:25 PM
1 min 41 sec
#11 puppet_production_deployment
Apr 25, 2014 3:00:11 PM
10 sec
#11 puppet_production_update_foreman
Apr 25, 2014 3:00:27 PM
14 sec
Pipeline #12
GITTAG : prod_20140423_1
#12 puppet_production_rspec_lint
Apr 23, 2014 4:06:33 PM
1 min 38 sec
#10 puppet_production_deployment
Apr 23, 2014 4:08:06 PM
6.9 sec
#10 puppet_production_update_foreman
Apr 23, 2014 4:08:18 PM
14 sec
Pipeline #11
GITTAG : prod_20140422_2
#11 puppet_production_rspec_lint
Apr 22, 2014 5:08:02 PM
1 min 37 sec
#9 puppet_production_deployment
Apr 22, 2014 5:09:44 PM
1.4 sec
#9 puppet_production_update_foreman
Apr 22, 2014 5:09:51 PM
13 sec
Pipeline #10
GITTAG : prod_20140422_1
#10 puppet_production_rspec_lint
Apr 22, 2014 11:49:49 AM
1 min 44 sec
#8 puppet_production_deployment
Apr 22, 2014 11:51:39 AM
6.5 sec
#8 puppet_production_update_foreman
Apr 22, 2014 11:51:51 AM
14 sec
Help us localize this page
Page generated: May 8, 2014 11:55:35 AM
REST API
Jenkins ver. 1.532.2

Monitoring


Monitor
#55 puppet_development_deployment 1s
#5 puppet_development_hieradata 0s
#3 puppet_development_hieradata_check 0s
#2797 puppet_development_rspec_lint 1m 35s
#51 puppet_development_update_foreman 57s
#2 puppet_production_deployment 5s
#1 puppet_production_hieradata 1s
#5 puppet_production_hieradata_check 0s
#3 puppet_production_rspec_lint 1m 5s
#3 puppet_production_rspec_lint_nodeploy 1m 35s
#2 puppet_production_update_foreman 17s
#13 puppet_testing_deployment 2s
#4 puppet_testing_hieradata 1s
#3 puppet_testing_hieradata_check 0s
#2769 puppet_testing_rspec_lint 1m 35s
#8 puppet_testing_rspec_lint_nodeploy 1m 36s
#5 puppet_testing_update_foreman 14s
Brought to you by Jan Molak

Wie verwalten wir Module von PuppetForge?

Puppetforge Module

- Eigenes GIT Repository (puppetforge.git)
- Download der Module in der Enwicklungsumgebung via `puppet module install ...`
- Staging GIT pull (bäh!)
- Dies ändert sich allerdings (dazu später)

- Wie soll eine Entwicklungsumgebung aussehen? *DONE*
- Wie testen wir den Puppet Code? *DONE*
- Wie verwalten wir unseren Puppet Code? *DONE*
- Wie soll unsere Puppet Umgebung aussehen? *DONE*
- Wie erfolgt das Deployment des Codes? *DONE*
- Wie verwalten wir Module von PuppetForge? *DONE*

Probleme, Probleme, Probleme...

- Ein GIT Repo funktioniert nicht bei Änderungen von Upstream Modulen
- Andere Abteilungen sollen ihre Module unabhängig testen
- Unittests sagen noch nichts aus, wie sich der Code am Live-System verhält
- Wir sollten eigentlich das Zusammenspiel aller Module testen (Forge und eigene)

Was haben wir geplant?

- r10k für Deployment (`https://github.com/adrienthebo/r10k`)
- Ein Repository pro Module
- Nur interne Module bleiben im Hauptrepo
- Acceptance Tests mit Beaker

Links und Bücher

- http://github.com/tosmi/linuxwochen2014
- http://github.com/tosmi/puppet-devel
- Puppet Learning VM: http://puppetlabs.com/download-learning-vm
- Foreman: http://theforeman.org
- Vagrant: http://vagrantup.com
- rspec-puppet: http://rspec-puppet.com/
- puppet-lint: http://puppet-lint.com/
- Gitolite: http://gitolite.com/
- r10k: https://github.com/adrienthebo/r10k
- Roles and Profiles: http://www.craigdunn.org/2012/05/239/
- Dynamische Puppet Umgebungen: http://puppetlabs.com/blog/git-workflow-and-puppet-environments
- puppet-sync: https://github.com/pdxcat/puppet-sync
- Instant Puppet 3 Starter
- Pro Puppet 2nd
- The RSpec Book
- Pulling Strings with Puppet
- Continuous Delivery

Danke für die Aufmerksamkeit

`http://stderr.at/foreman_puppet_development2014.pdf`